Name: Mukta Akter Mim   15/09/21

ID: 2018-3-60-009

Course: CSE 246.

Ans: to the Q/A No. 2
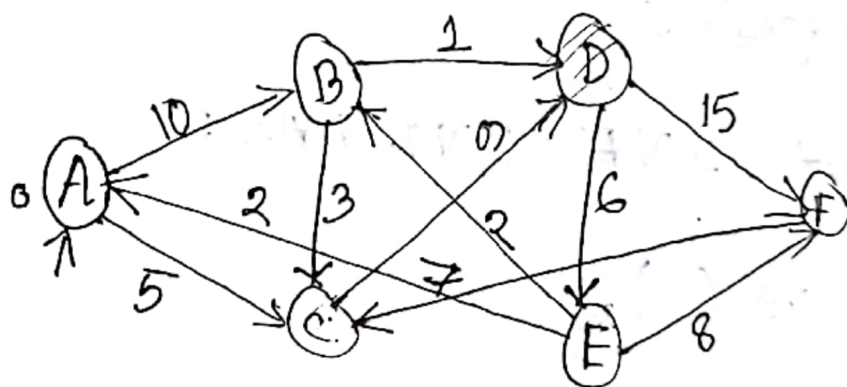
Vertex	A	B	C	D	E	F
A	[0]	∞	∞	∞	∞.	∞
C		10	[5]	∞	∞	∞
B		[10]		14	∞	∞
D				[11]	∞	∞
E					[17]	26
F						[25]

```
struct edge
{
    int a, b, cost;
};

int m, n, v;
vector <edge>;
const int INF = INT_ MAX;
void solve ()
{
    vector <int> d (n, INF);
    d[v] = 0;
    for (i=0; i<n-1; i++)
        for (j=0; j<m; j++)
            if ( d[e[j].a] < INF )
```

$$d[e[j].b] = \min (d[e[j].b],$$
$$d[e[j].a] +$$
$$e[j].cost);$$

Here, few phases are useful work is done, other remaining phases are useless. So, let's keep a flag variable.

int f = true;

f will check whether something changed the current phase or not. This accelerate the behavior of the algorithm.
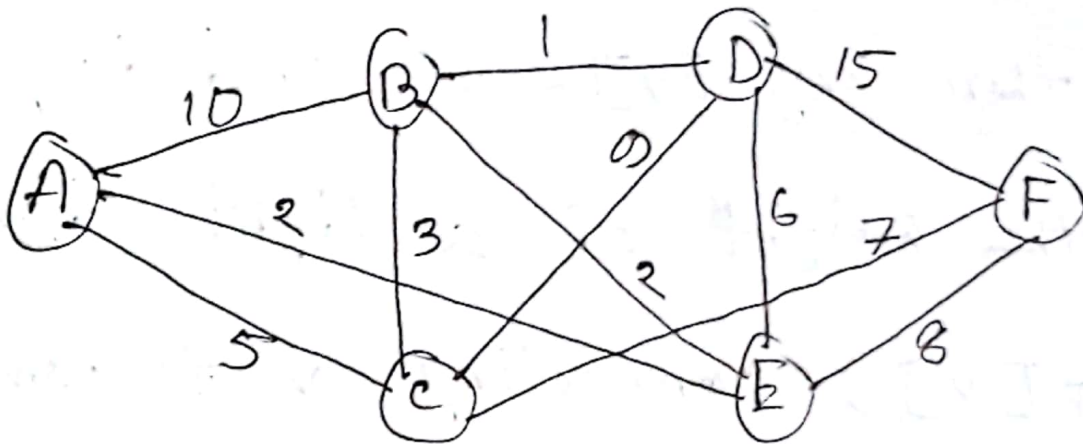
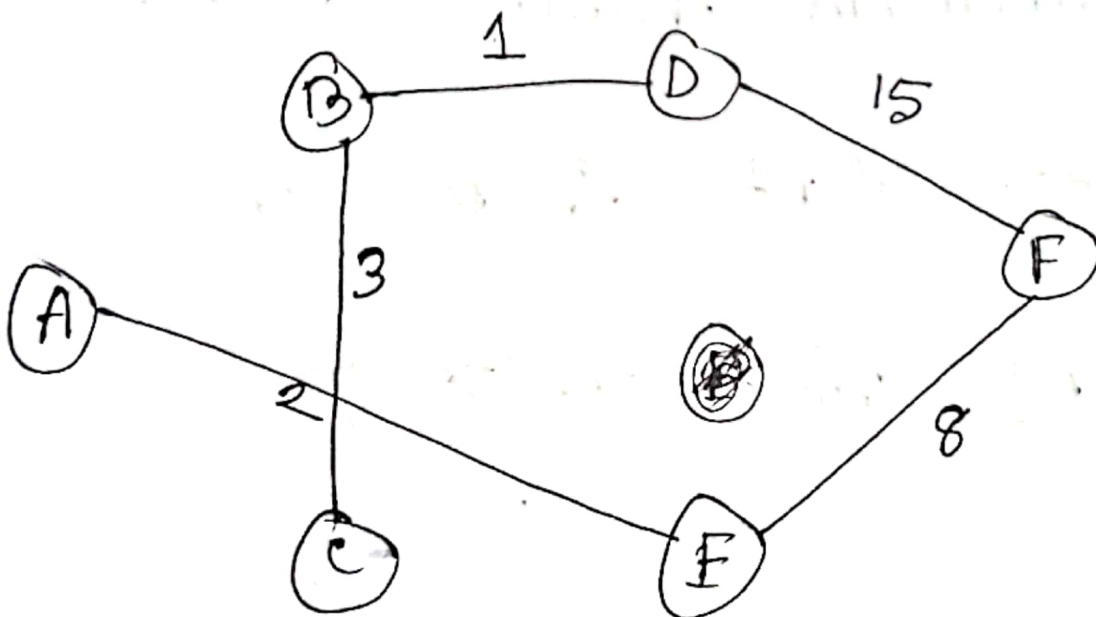So, bool f = false.

if (cost change)
   f = true;
if (!f)
      break;

From Question 2,

Kruskal:

For all vertices $V \in G$, if $\pi[v]$ nil, then key $[v] < \alpha$. and d key $[v]$ is the weight of a light edge $(v, \pi[v])$ connecting v to some vertex already pinched into the minimum spaning tree.

The tree stants from an arbitary root vertex T.

D^0

0	3	5	int	int	int
int	0	9	15	int	int
int	int	0	9	int	int
int	int	int	0	9	11
int	int	int	int	0	3
int	int	int	int	int	0

D^1

0	3	5	int	int	int
int	0	9	5	int	int
int	int	0	9	int	int
int	int	int	0	7	11
int	17	~~int~~20	int	0	3
int	int	11	int	int	0